Abhisek Acharya

Comparative Study of Machine Learning Algorithms for Heart Disease Prediction

Helsinki Metropolia University of Applied Sciences

Information Technology

Thesis

28 April 2017

Abstract

Author(s) Title	Abhisek Acharya Comparative study of machine learning algorithms for heart disease prediction
Number of Pages Date	43 pages + 4 appendices 4 Nov 2016
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Sakari Lukkarinen, Senior Lecturer

As technology and hardware capability advance machine learning is also advancing and the use of it is growing in every field from stock analysis to medical image processing. Heart disease prediction is one of the fields where machine learning can be implemented. Therefore, this study investigates the different machine learning algorithms and compares the results using different performance metrics i.e., accuracy, precision, recall, f1-score etc. The dataset used for this study was taken from UCI machine learning repository, titled “Heart Disease Data Set”.

This study was executed as a quantitative case study and several previous research on this data set was studied and analysed deeply to understand the subject in greater depth. Statistics and numbers are widely used in the study, so that the study can be quantifiable and several correlations between data attributes can be found.

The main objective of this study was to compare the algorithms which can classify the heart disease correctly based on different performance metrics. There are 13 dependent variables in the data set and 1 independent variable to be predicted. The original data set contains predicted variables from 0 to 4 representing a healthy heart starting from 0 to severely unhealthy heart at 4. For this study, 0 to 4 class labels were changed to 0 and 1. The predicted class can be either 0 or 1, meaning the heart is either 0 (“Healthy”) or 1 (“Unhealthy”). Techniques such as feature selection, grid search and probability calibration were used to get the optimal results.

In this study, algorithms such as k-Nearest Neighbour, Support Vector Machine, Naïve Bayes, Adaboost, Random Forest and Artificial Neural Network are used. It can be concluded that Artificial Neural Network and Support Vector Machine are best the algorithms for this data set and possibly other heart disease data sets. For the proper conclusion for this study to be applied clinically, it needs to be further elaborated with the help of experts in both heart and machine leaning domains.

Keywords	heart, classification, SVC, ANN, Naïve Bayes, training set

List of Abbreviations

Ada Boost	Adaptive Boosting
ANN	Artificial Neural Network
CVD	Cardiovascular Disease
FP	False Positive
FN	False Negative
KNN	k-Nearest Neighbour
ML	Machine Learning
NB	Naïve Bayes
RF	Random Forest
SBS	Sequential Backward Selection
SFS	Sequential Forward Selection
SVC	Support Vector Classifier
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
UCI	University of California

Contents

1 Introduction 1

2 Theoretical Background 2

2.1 Overview of Machine Learning 2
2.2 Dimension Reduction Techniques 3
2.2.1 Feature Selection 3
2.2.2 Feature Extraction 5
2.3 Machine Learning Algorithms 6
2.3.1 k-Nearest Neighbour 6
2.3.2 Random Forest 8
2.3.3 Support Vector Machine (SVM) 9
2.3.4 Naïve Bayes 10
2.3.5 Ada Boost 10
2.3.6 Artificial Neural Network 12
2.4 Performance Metrics 13
2.6 Probability Calibration 17
2.7 Data Pre-processing 19

3 Methods and Materials 20

3.1 Research Approach 20
3.2 Research Design 21
3.3 Data Collection and Description 22
3.4 Tools Used 23
3.5 Data Cleaning and Scaling 25
3.6 Data Reduction 25
3.7 Descriptive Statistics and Exploratory Data Analysis 25

4 Results and Discussion 30

4.1 Overview **Error! Bookmark not defined.**
4.2 Experiments 30
4.2.1 Training Set 30
4.2.2 Test Set 31
4.2.3 Results 31
4.3 Comparison of Results 38

5 Conclusion 39

References 40

Appendices

Appendix 1. Sample of Data

Appendix 2. Sample of OneHotEncoded Data

Appendix 3. Sample code of Naïve Bayes with Feature Selection

1 Introduction

Cardiovascular disease (CVD) is increasing day by day in this modern world. According to WHO, an estimated 17 million people die of CVDs particularly heart attack and strokes, every year [1]. Thus, it is necessary to enlist the most important symptoms and health habits which contribute towards CVDs.

Different tests are conducted before diagnosing CVD, Some of them are auscultation, ECG, blood pressure, cholesterol and blood sugar. These tests are often extensive and time consuming while a patient's health condition may be critical and he/she needs to start immediate medication so it becomes important to prioritize the tests. There are several health habits which contribute to CVDs. Therefore, it is also necessary to know which of the health habits contribute to CVDs so that healthy health habits can be practiced.

Machine learning is an emerging field today due to a rise in the amount of data. Machine learning helps to gain insight from a massive amount of data which is very cumbersome to humans and sometimes also impossible. The study's objective is to prioritize the diagnosis test and see some of the health habits which contribute to CVD. Furthermore, and most importantly, different machine learning algorithms are compared according to different performance metrics. In this thesis, manually classified data is used. Manual classification is either healthy or unhealthy. Based on a machine learning technique called classification, 70 percent data are supervised or trained and 30 percent are tested in this thesis. Thus, different algorithms are compared as per their prediction results.

2 Theoretical Background

2.1 Overview of Machine Learning

Machine learning is a subfield of computer science and a rapidly up surging topic in today's context and is expected to boom more in coming days. Our world is flooded with data and data is being created rapidly every day all around the world. According to Big Data and Analytics Solutions company CSC, it is expected by 2020, that the data amount will be 44 times bigger than in 2009 [2]. Therefore, it is necessary to understand data and gain insights for better understanding of a human world. The data amount is so huge today that traditional methods cannot be used. Analysing data or building predictive models manually is almost impossible in some scenarios and also time consuming and less productive. Machine learning, on other hand, produces reliable, repeatable results and learns from earlier computation.

Data used for machine learning are basically of two types labelled data and unlabelled data. Labelled data is the data where attributes are provided. It has some sort of tag or meaning attached to the data therefore used in supervised learning. Labelled attribute can be numerical or categorical. Numerical data are used in regression to predict the value while categorical data are used in classification. Unlabelled data is the data where there are only data points and no labelling to assist. Unlabelled data are used in unsupervised learning so that machine can identify the patterns or any structure present in the data set. [3, 3]

The labelled data and unlabelled data are used with supervised learning and unsupervised learning respectively. Supervised learning entails a learning map between a set of input variables X and an output variable Y and applying this mapping to predict the output for unseen data [3]. After learning the dataset, algorithms generalise the data and formulates the hypothetical value H for the given dataset.

Supervised learning is further categorised into two types: Regression and Classification. According to the business dictionary, a regression is a technique for determining the statistical relationship between two or more variables where a change in dependent variable is associated with, and depends on a change in one or more independent variables [4]. Classification is a task that occurs very frequently in everyday life. Essentially it involves dividing up objects so that each is assigned to one of a number of mutually exhaustive and exclusive categories known as classes. The term ‘mutually exhaustive and exclusive’ simply means that each object must be assigned to precisely one class, i.e. never to more than one and never to no class at all [5,5].

Unsupervised learning studies how systems can learn to represent particular input patterns in a way that reflects the statistical structure of the overall collection of input patterns. By contrast, with supervised learning there are no explicit target outputs or environmental evaluations associated with each input; rather the unsupervised learner brings prior biases as to what aspects of the structure of the input should be captured in the output. [6, 858]

2.2 Dimension Reduction Techniques

Dimensionality reduction is simply the process of decreasing the number of input random variables without the loss of information. A greater number of input variables or dimensions and large data samples increase the complexity of the dataset. To reduce the memory and computational time dimensionality of data is reduced. Dimensionality reduction also helps to eliminate unnecessary input variables like duplicate variables or variables with a very low significance level. [7, 109-110]

There are two types of dimensionality reduction techniques: Feature Selection and Feature Extraction are described below.

2.2.1 Feature Selection

In feature selection, k dimensions are selected out of d dimensions that gives most information and discard the (d-k) dimensions. In other words, feature selection is also called subset selection. The best subset contains the least number of dimensions that contribute most to the accuracy. The best subset is found with suitable error function. [7, 110.]

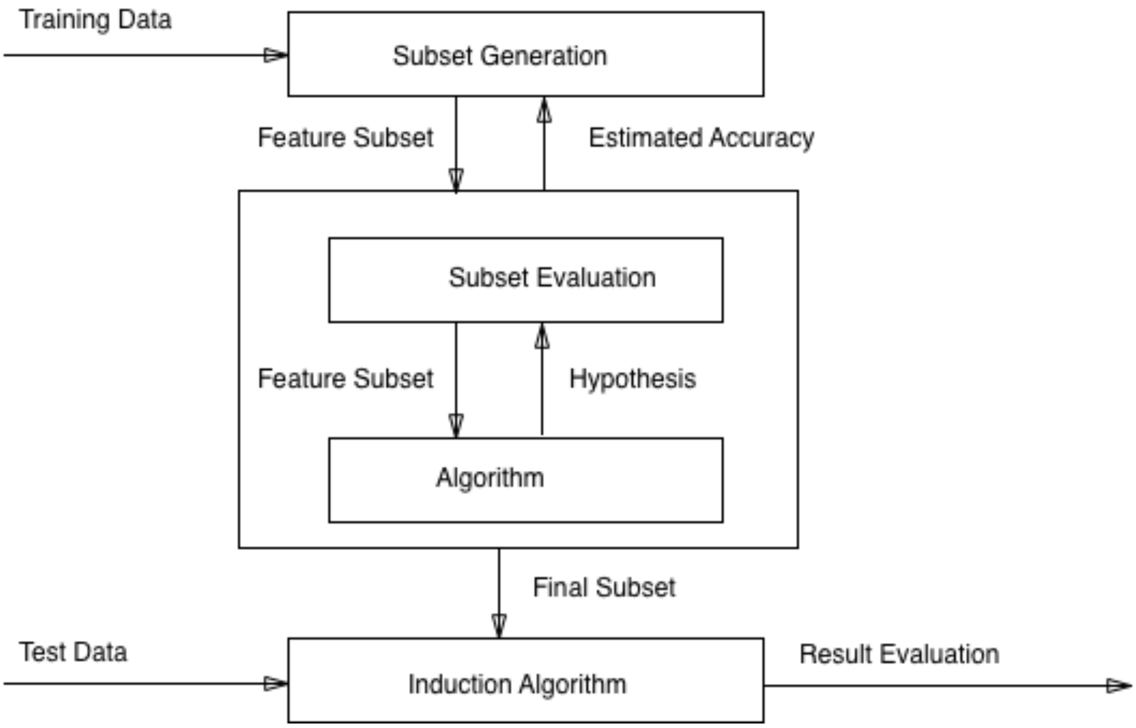


Figure 1 Feature Selection. Adapted from Md Rahat Hossian (2013) [8]

In Figure 1, Training data is put through a certain process of subset generation, for example sequential backward selection. The resulting subset is now put through the algorithm to test its performance if the performance meets the expected criteria, then it will be selected as the final subset. Otherwise, the resulting subset will again be put through the process of subset generation for more fine-tuning.

There are two different approaches to feature selection: Sequential Forward Selection and Sequential Backward Selection which are explained below.

Sequential Forward Selection

Sequential Forward Selection begins with a model containing no predictors, and then predictors are added to the model, one at a time, until all of the predictors are in the model. In particular, at each step the variable that gives the greatest additional improvement to the fit is added to the model [9,207].

Let us denote a set by *P*, with variables *Xi, i = 1,……,d. E(P)* is the error incurred in the test sample. Sequential Forward Selection starts with empty set with no variables *P* = {*ϕ*}. At each step, a single variable is added to the empty set and a model is trained, and error *E(P∪ Xi)* is calculated on test set. Error criteria is set as per requirement, for example, the least square error and misclassification error. From all the errors, the input variable causing the least error X*j*, is selected and added to the empty set *P*. The model is trained again with remaining number of variables and the process continues to add variables to *P*, if *E(P∪ Xi)* is less than *E(P)*.

Sequential Backward Selection

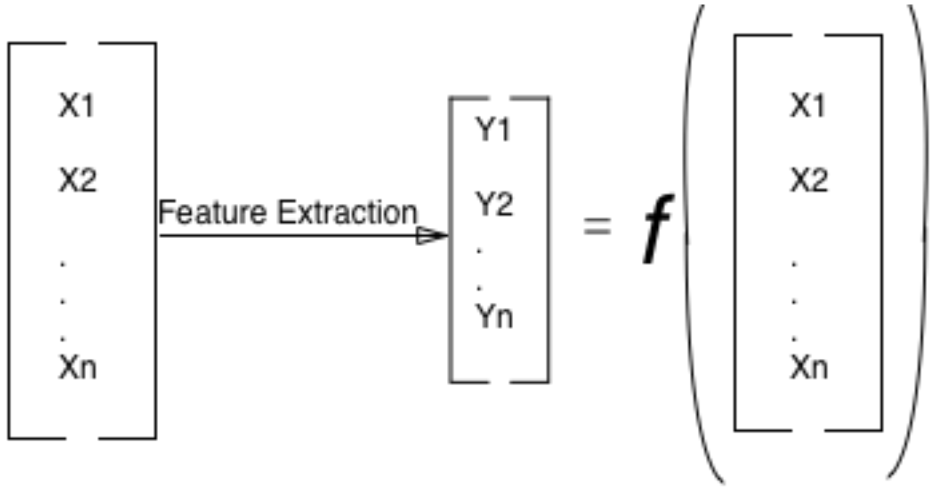
Sequential Backward Selection is an efficient alternative for best subset solution but it begins with full set of features unlike Sequential Forward Selection. It removes the least significant features one at a time iteratively. [9,207.]

Sequential Backward Selection starts with a full set of variables *P = {1,2,3,…..,d}*. At each step, the model is trained with a full set of variables and the error is calculated in test set, and the variable with highest error X*j*, is removed from the set *P*. The model is trained again with a new set of variables *P*, and the process continues to remove variables from *P*, if *E(P-Xj)* is less than *E(P)*.

2.2.2 Feature Extraction

In the feature extraction technique, features or independent variables from the data set are transformed into new independent variables known as new feature space. Newly constructed feature space explains the data most and only significant data are selected.

Let, there are n features of X_1…….X_n. After feature extraction there are m attributes where (n > m) and this feature extraction is done with some mapping function, F.



Referring to Figure 2, Xn set of independent features or dimensions are reduced to Yn set of independent features. In the feature extraction process, a technique such as principal component analysis is used. It will take only non-redundant and significant features from Xn and change into new feature space Yn [11]. With the feature extraction, capability of interpretation is lost since, Yn features obtained after feature extraction is not same as Xn meaning it is not a direct subset of Xn.

2.3 Machine Learning Algorithms

For the purpose of comparative analysis, six Machine Learning algorithms are discussed. The different Machine Learning (ML) algorithms are k-Nearest Neighbour (KNN), Random Forest (RF), Support Vector Machine (SVM), AdaBoost, Naïve Bayes and Artificial Neural Network (ANN). The reason to choose these algorithms is based on their popularity [12].

2.3.1 k-Nearest Neighbour

Nearest Neighbour algorithms are among the simplest of all machine learning algorithms. The idea is to memorize the training set and then to predict the label of any new instance on the basis of the labels of its closest neighbours in the training set. The rationale behind such a method is based on the assumption that the features that are used to describe the domain points are relevant to their labelling in a way that makes close-by points likely to have the same label. Furthermore, in some situations, even when the training set is immense, finding a nearest neighbour can be done extremely fast. [13,258.]

Mathematically, $\rho\colon X * X \;\to\; \mathbb{R}$ where Ψ is a function that returns the distance between the two points of $X\ (x_i, x_i')$. The Euclidean distance between two points can be calculated by following formula:

$$\rho(x, x') = |x - x'| = \sqrt{\sum\nolimits_{i=1}^{d} (x_i - x'_i)^2} \qquad (\,2.1)$$

k in the k-nearest neighbour is the number of the data points closest to the new instance. For example, if k =1 then the algorithm will choose the nearest one instance or if k = 4, then the algorithm will choose the closest four neighbour instances and will classify them accordingly. The idea can be better illustrated with figure 3.

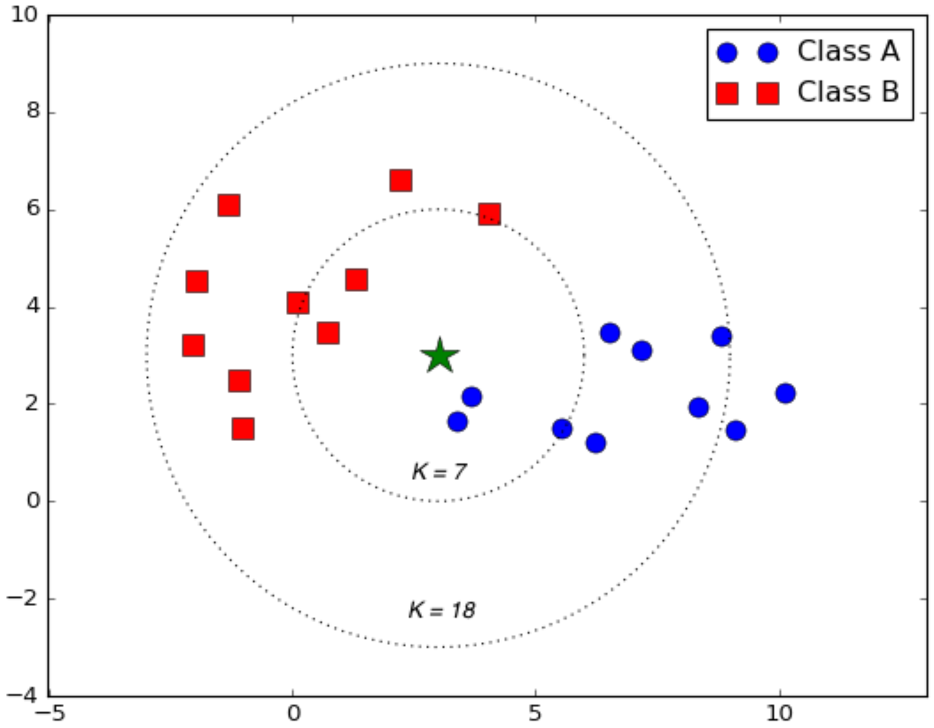


Figure 3 k- Nearest Neighbour. Modified from K Nearest Neighbour and Dynamic Time Wrapping (2016) [14]

Referring to Figure 3, Green star is the data point to be classified, blue circles are class A data points and red squares are class B rectangles. The Euclidean distance between the green star and all other red and blue points are measured. The star will be classified to the data points which have least distance. If k =7, then distance between all the seven points are measured from the star and star will be classified to the data points with least distance, in this case with blue data point.

2.3.2 Random Forest

A random forest is a classifier consisting of a collection of decision trees, where each tree is constructed by applying an algorithm A on the training set S and an additional random vector, θ where θ is sampled i.i.d.(independently and identically distributed) from some distribution. The prediction of the random forest is obtained by a majority vote over the predictions of the individual trees [13,255].

The Random Forest algorithm works in following steps:

1. Picks random K data points from the training data.
2. Builds a decision tree for these K data points.
3. Chooses the Ntree subset from the trees and performs step 1 and step 2
4. Decides the category or result on the basis of the majority of votes.

To understand Random Forest more intuitively it is better to understand decision trees and they can be better understood with the help of a diagram.

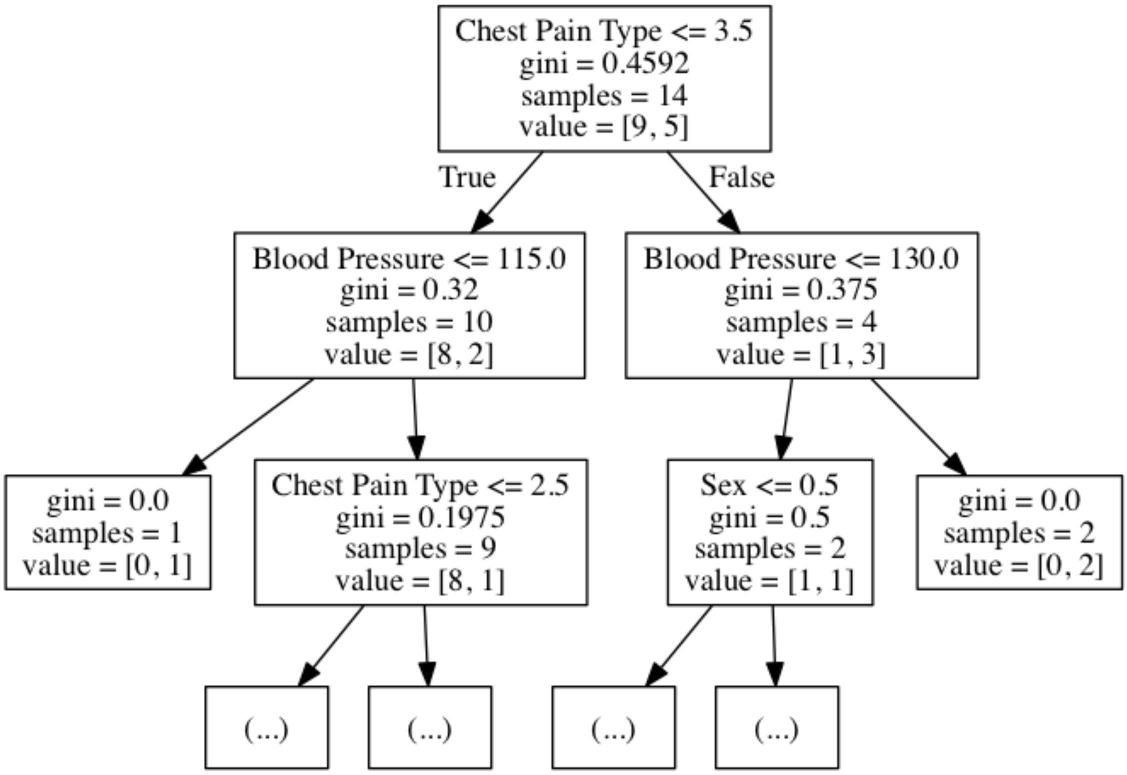


Figure 4 Random Forest Schematic.

Referring to Figure 4, it illustrates how decision trees work. If decision trees are used to predict whether there is a heart disease or not, then it will decide according to the above mapping. Gini is the coefficient of the spread of the data, samples are the number of data taken for classifying the node, value is the array of samples which are classified as true or false.

2.3.3 Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a classifier which distinct the various classes of data by the use of a hyper-plane. SVM is modelled with the training data and it outputs the hyper-plane in the test data. [15]. The SVM model tries to find the space in the matrix of data where different classes of data can be widely differentiated and draws a hyper-plane.

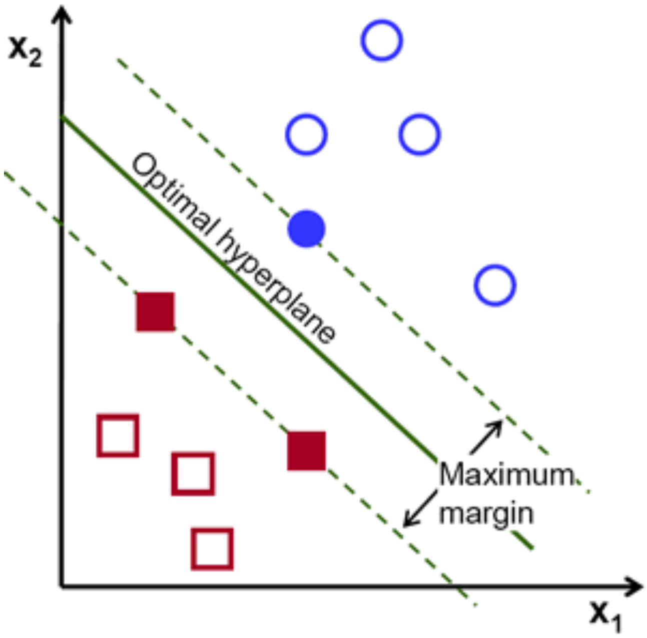


Figure 5 Support Vector Machine. Reprinted from Introduction to SVM [15]

In Figure 5, Red and Blue are the classes of labelled training data points. To classify them linearly a hyper-plane can be drawn but the question is: There is more than one way to draw a hyper-plane so which one is optimal? An optimal hyper-plane is chosen which maximizes the margin between the classes. Hyper-plane need not always be linear. A hyper-plane in SVM can also work as a non-linear classifier using technique known as kernel-trick.

2.3.4 Naïve Bayes

Naïve Bayes or Naïve Bayes classifier in a machine learning context is a classifier which uses the Bayes theorem to classify the data and it assumes that the probability of certain feature X is totally independent of another feature Y [7,397]. Bayes theorem can be easily explained with the following example.

Probability of spanners produced by machine A is 0.6, machine B is 0.4. A defect in spanners in the whole of production is 1 percent and the probability of defected spanners produced by machine A is 50 percent and machine B is 50 percent. In this scenario Bayes theorem can be used to answer what is the probability of a defected machine produced by machine B is ?

$$P(Defect \mid Machine\ B) = \frac{P(Machine\ B \mid Defect) * P(Defect)}{P(Machine\ B)} \qquad (2.2)$$

Bayes theorem provides a way to calculate the probability of the hypothesis given that there is prior knowledge about the problem.

$$Posterior = \frac{Likelihood * Prior}{Evidence} \qquad (2.3)$$

There are three types of Naïve Bayes. Gaussian Naïve Bayes, Multinomial Naïve Bayes and Bernoulli Naïve Bayes. Gaussian Naïve Bayes is used in classification problems, Multinomial Naïve Bayes is used in multinomial distributed data and Bernoulli Naïve Bayes is used in data with multivariate Bernoulli distribution.

2.3.5 Ada Boost

Ada Boost is the one of the robust techniques proposed by Freund and Schapire and most practical and widely used boosting algorithm [16]. Boosting is an ensemble technique in machine learning used for creating highly accurate predictions or strong classifier from relatively weak and inaccurate classifiers. Boosting is an iterative process where a model is trained in data and finds weak learners. The second model is trained in data which learns from the mistakes of previous training and fixes the errors and the process continues until the training data is correctly predicted.

Let the number of training samples be 20 in a binary classification problem and training samples be uniformly distributed, D_1, initial weight of each sample *w* is *1/20*. After the training samples are trained in a model G_1, 5 of them are misclassified. Therefore, error rate *e* = *0.25*, the weight of model G_1 is

$$G_1 : \alpha_1 = \frac{1}{2}\ln\left(\frac{1-e_1}{e_1}\right) \tag{2.4}$$

Misclassified samples are weighted more by multiplying with e^{α_1} and correctly classified samples are weighted less by multiplying with $e^{-\alpha_1}$ and all weights are normalized to 1. Now, the second model is trained G_2 and the same process continues until optimum samples are correctly classified and final model is the sum of all weighted models as shown in Figure 6.

Schematic of Adaboost is illustrated below.

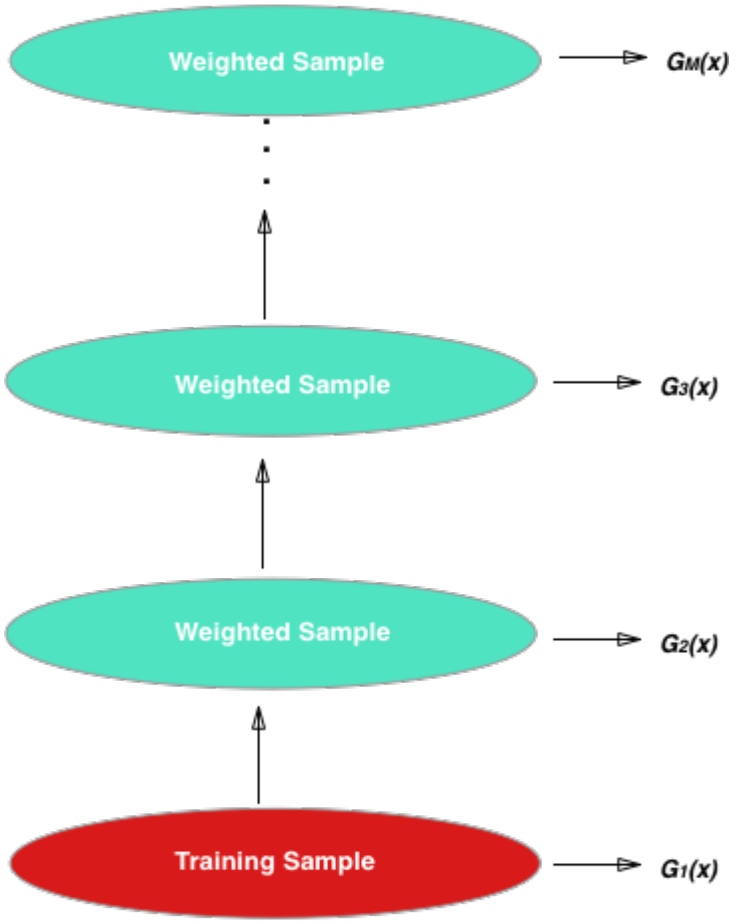


Figure 6 Adaboost Schematics. Adapted from Trevor Hastie (2001) [17,338]

In Figure 6, training sample is the first set of independent predictor variables which are to be modelled. After the modelling error rate is achieved and weak learners are identified. These newly achieved weak learners are weighted and weighted samples are again modelled and this process continues up to M number of times and finally all the outputs from different samples are averaged to give boosted output.

2.3.6 Artificial Neural Network

Artificial Neural Network is modelled after biological neural networks and attempts to allow computers to learn in manners similar to human-reinforcement learning [18]. The basic unit in neural network which operates in known as perceptron. A perceptron consists of one or more inputs, a processor, and a single output. A perceptron follows the "feed-forward" model, meaning the inputs are sent to a neuron, processed and result in output.

A perceptron follows four main steps.

1. Receive inputs
2. Weight inputs
3. Sum inputs
4. Generate inputs

Each input that is sent into the neuron must first be weighted, i.e. multiplied by some value (often a number between -1 and 1). Creating a perceptron typically begins by assigning random weights. Each input is taken and multiplied by its weight. The output of the perceptron is generated by passing that sum through an activation function. In the case of a simple binary output, the activation function is what tells the perceptron whether to "fire" or not. Many activation functions to choose from (Logistic, Trigonometric, Step etc.). For example, let us make the activation function the sign of sum. In other words, if the sum is positive number, the output is 1; if it is negative, the output is -1. One more factor to consider is bias. Imagine both inputs were equal to zero, then any sum no matter what multiplicative weight would also be zero.

To avoid the problem, a third input is added known as bias input with a value of 1. This avoids the zero issue.

To actually train the perceptron following steps are used:

1. Provide the perceptron with inputs for which there is a known answer.
2. Ask the perceptron guess for an answer.
3. Compute the error (How far off from the correct answer?)
4. Adjust all the weights according to the error, return to step 1 and repeat.

We repeat this until we reach the error we are satisfied with. That is how a single perceptron would work. Now to create a neural network all that is needed is to link many perceptrons together in layers, input layer and output layer. Any layers in between input and output layer are known as hidden layers.

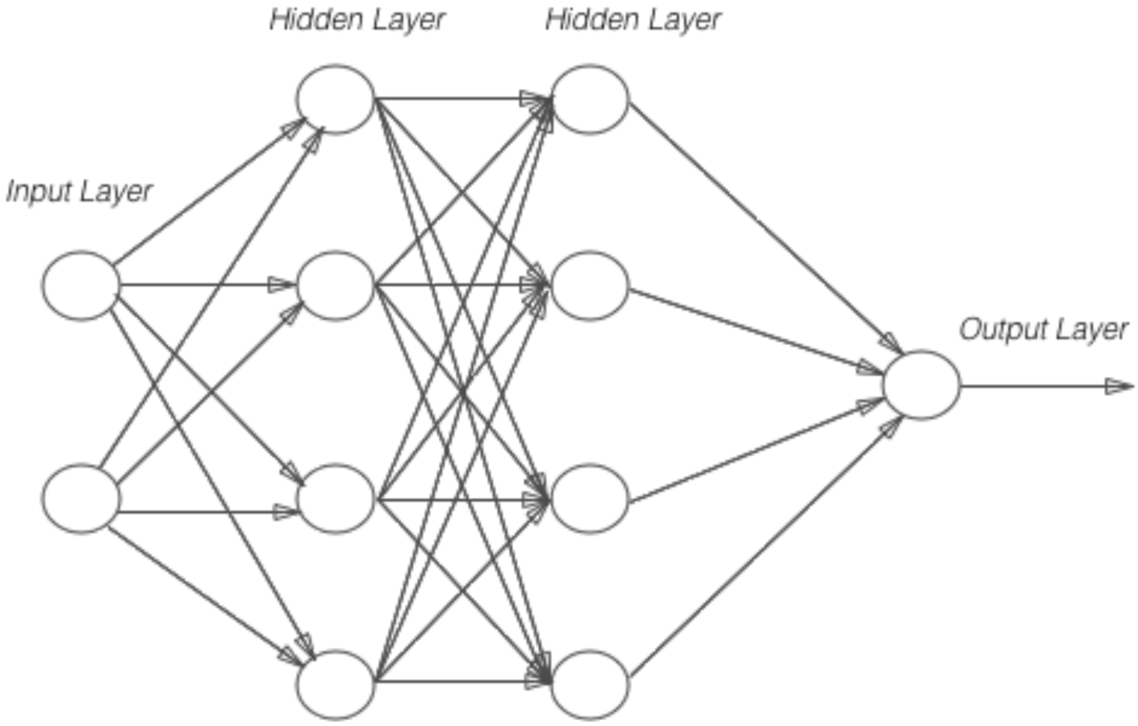


Figure 7 Schematic of Artificial Neural Network

In Figure 7, a perceptron consists of one input layer, two hidden layer and one output layer.

2.4 Performance Metrics

In a machine learning context, performance metrics is the measurement of algorithm on how well algorithm performs based on different criteria such as accuracy, precision, recall etc. Different performance metrics are discussed below.

Confusion Matrix

Confusion matrix is a technique to show how the classifier is confused while predicting. In a binary classification problem, the confusion matrix is often illustrated as below.

Table 1. Confusion Matrix

Correct classification	Classified as	
	+ (1)	- (0)
+ (1)	TP (1,1)	FN (1,0)
- (0)	FP(0, 1)	TN(0,0)

In Table 1, TP is the true positive value which means the positive value is truly classified, FP is the false positive value which means positive value which is falsely classified, FN is the false negative value which means negative value which is falsely classified and TN is the true negative which means negative value which is truly classified.

From the confusion matrix, different performance metrics can be calculated. Using Table 1 as an example, accuracy, precision, recall and F1-score are explained in more detail below.

Accuracy

Accuracy or predictive accuracy is the measure of the proportion of instances that are correctly classified [5, 178]. It shows how close the predicted value is to the true value or the theoretical value. Formula for calculating the accuracy is given below.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \qquad (2.6)$$

Accuracy is the measure of how close or near the predicted value is to the actual or theoretical value. For example, if the actual value of a person’s height is 6 feet and measured value or predicted value is 5.9 then it is quite an accurate measurement.

Precision

Precision is defined as the proportion of true positive instances which are classified as positive [5, 178]. It shows how close predicted values are to each other. Formula for calculating precision is given below.

$$Precision = \frac{TP}{TP + FP} \qquad (2.7)$$

Precision is the measure of how close or near the predicted values are to each other. For example, if actual value of person's height is 6 feet and measured value is or predicted value is 5.5 and every time measurement is taken height is 5.4 or 5.6 then prediction is quite precise but not accurate.

Recall

Recall is defined as the proportion of positive instances are that correctly classified as positive [5,178]. Recall is also often called sensitivity. Formula for calculating recall is given below.

$$Recall = \frac{TP}{TP + FN} \qquad (2.8)$$

Recall is simply the measure of how many true samples are predicted from the all the samples.

F1 Score

F1 score is defined as the measure which combines both precision and recall and tries to convey the balance between them. Formula for calculating F1 score is given below.

$$F1\,Score = \frac{2 * Precison * Recall}{Precision + Recall} \qquad (2.9)$$

F1 score shows how good the classifier is in the context of both precision and recall. Therefore, it is good to have both precision and recall value be more to have better F1 score. Either precision or recall value is low than the overall F1 score decreases.

ROC Curve

ROC Curve, Receiver Operating Characteristic is a graphical way to show how good the performance of a classifier is. Basically, it is a plot of a true positive rate against a false positive rate.

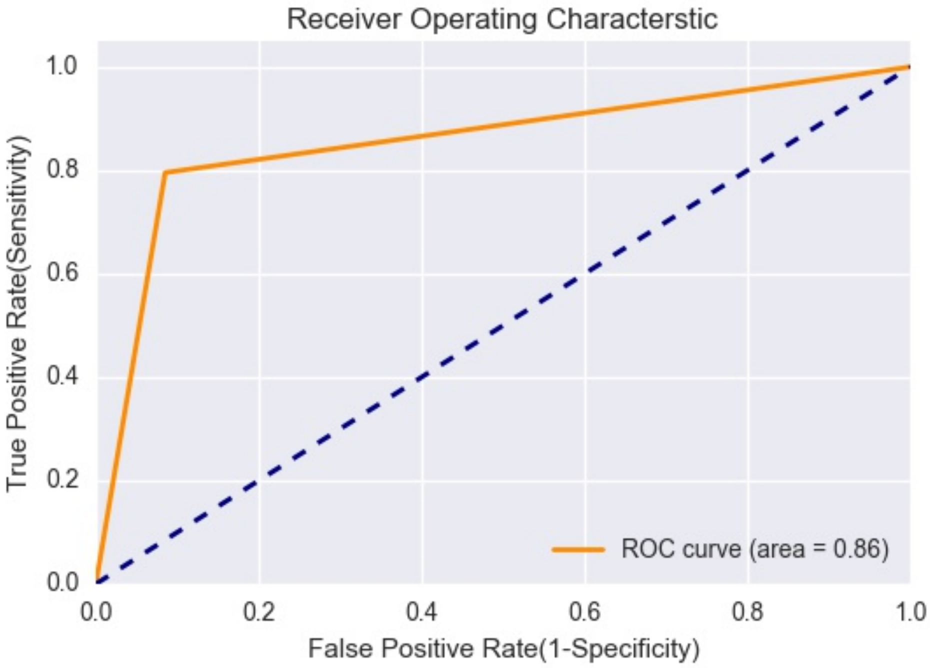


Figure 8 ROC Curve

In Figure 8, orange line represents the true positive rate the classifier has achieved, the dotted line is a random line which classifies the data in two halves. When the orange line goes below the dotted lines it means the classifier is worse than random guessing and as the orange line goes above the classifier is good. If orange line reaches the value of 1 in Y-axis than that is perfect classifier.

Area covered below the orange line is called Area Under the Curve, AUC, which means, larger the area under covered by the curve better is the classifier. Maximum value is 1 and anything more than 0.90 is considered excellent. Simultaneously, 80 to 90, 70 to 80, 60 to 70 and 50 to 60 are good, fair, poor and fail [19].

Cross Entropy

Cross Entropy is the name commonly used when the classification is not binary and a log loss is used when classification is binary. Underlying mathematics for both cross entropy and log loss is same. A log loss is defined as “negative log-likelihood of the true labels given a probabilistic classifier’s prediction” [20]. Log loss quantifies the accuracy of classifier by penalising false classification. It means log loss maximizes the accuracy of classifier, therefore lower the log loss better is the classifier. Log loss is often used in competition like Kaggle to evaluate the accuracy of the model. Formula for log loss is given below.

$$Log\ loss\ (L) = -\frac{1}{n}\sum\nolimits_{i=1}^{n}[y_i\, log(\widehat{y_i}) + (1-y_i)\log(1-\widehat{y_i})] \qquad (2.10)$$

For each row in the dataset, y_i is the outcome or the dependent variable whose value is either 0 or 1. $\widehat{y}_i$ is the predicted probability outcome obtained by applying the logistic regression equation. The objective is to adjust the estimates in the logistic regression equation such that the total log loss function over the whole dataset is minimized and if y_i is 1, then the function is minimized with the high value of $\widehat{y}_i$ and if y_i is 0, then the function is minimized with a low value of $\widehat{y}_i$.

2.6 Probability Calibration

Probability calibration is a classifier which outputs class probabilities rather than the class labels. To have some confidence in the predicted output it is sometime better to know the probability of the predicted class. However not all the classifiers do support the probability output and some of the gives poor results in probability output. So before comparing the classifiers, it is better to calibrate.[21]

There exists a special curve named as reliability curve which helps to find out which of the classifiers need to be calibrated. In the reliability curve, classifier having a plot near to diagonal is well calibrated and vice-versa. Classifiers which are way off the diagonal line need to be calibrated.

There are two types of calibration used in probability calibration named Platt Scaling and Isotonic Regression. These two methods are designed for binary classification and it does not work for multiclass problems. One way to use these methods in multiclass problem is to change multiclass problems into binary classification. Platt calibration is simply obtained by passing the classifier through special function called sigmoid function and Isotonic calibration is obtained by passing the predicted output or mapping the predicted output through monotonically increasing function.

Figure 9 shows the reliability curve and mean predicted value of different classifiers.

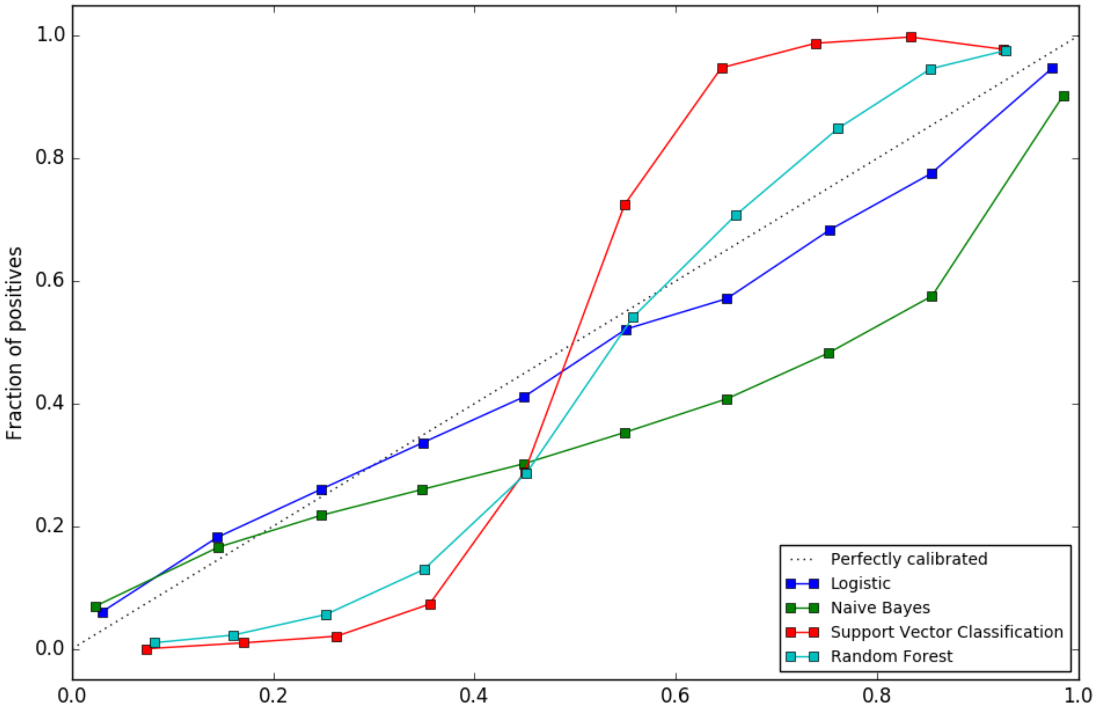


Figure 9 Reliability Curve. Reprinted from Probability calibration [21]

In the reliability diagram above, the logistic regression curve is very close to perfectly calibrated line which is a diagonal. It means logistic classifier needs not be calibrated while the Support Vector Machine curve is distorted from diagonal, it means SVM gives very poor estimates of the predicted probabilities and needs to be calibrated.

2.7 Data Pre-processing

In the modern world, there is a large amount of data collected through various sources such as the internet, surveys and experiments etc. But most of the time the data to be used are full of missing values, noises and distortions. Since, majority of data collected are not taken from controlled environment, the data can contain garbage in it. "Garbage in – Garbage out" principles work here also. No matter how well data is modelled, if data contains garbage then output obtained is always garbage. Therefore, data pre-processing is a fundamental step to data analysis and machine learning. Data pre-processing or often called data wrangling is a set of procedures which transforms raw data into understandable format.

Data pre-processing includes detecting outliers, making decisions what to do with the outliers, finding and filling appropriate missing values and searching for inconsistency in the data. Data needs to normalized or standardized and reduced before applying to machine learning algorithms. Normalization of data is done when the features of dataset have different measuring units. For example, Fahrenheit and Celsius, though both are units of temperature their measuring units are different. Standardization is used to scale the data and it gives the information on how many standard deviations the data is from its mean value. Standardization rescales the data to have the mean (μ) of 0 and the standard deviation (σ) of 1.

The formula for standardization is given below.

$$Standardization\left(\overline{X}\right) = \frac{X - \mu}{\sigma} \qquad (2.11)$$

Data is reduced, also known as feature selection when there are redundant features in the dataset or the features are very insignificant for the results.

3 Methods and Materials

This section describes the different methods and materials used for this study i.e., research approach, research design and implementation, data collection and tools used for this study.

3.1 Research Approach

This thesis examines the empirical relationship between the set of features such as age, sex and blood pressure with the probability of being diagnosed with heart disease Therefore, this thesis is a quantitative case study.
Researcher Robert K. Yin defines “a case study as an empirical inquiry that investigates a contemporary phenomenon within its real-life context; when the boundaries between phenomenon and context are not clearly evident; and in which multiple sources of evidence are used [22].”

Quantitative case study approach [23, and 22] is chosen in this thesis because the idea of this study is to investigate available heart disease datasets using a number of statistical methods and running machine learning algorithms on them, so as to find out which one of the machine learning algorithms give better results. Similar studies have been done in the past. The motive of this study is to replicate and extend the previous studies.

3.2 Research Design

This section illustrates the research design; it describes the actual flow the of the entire research.

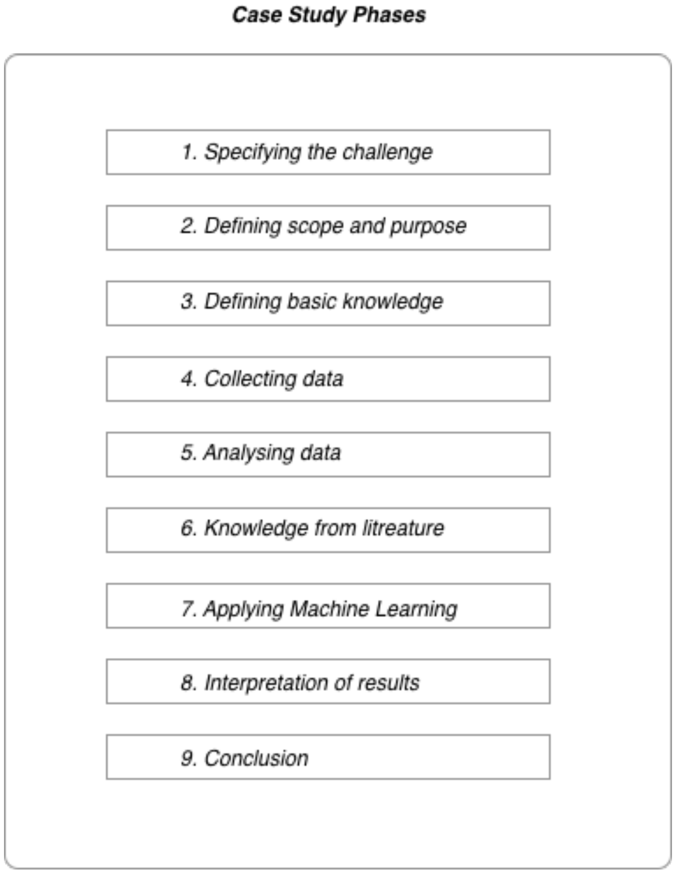

Case Study Phases
1. Specifying the challenge
2. Defining scope and purpose
3. Defining basic knowledge
4. Collecting data
5. Analysing data
6. Knowledge from litreature
7. Applying Machine Learning
8. Interpretation of results
9. Conclusion

Figure 10 Flow of the case study

The first step of this study was to use an available heart disease data set and compare different machine learning algorithms to understand their performance based on different performance metrics. In the second step, scope for the study and its purpose was defined. The scope of the study was defined as per study purpose, resources and schedule. The studies purpose was to understand the machine learning algorithms behaviour in particular heart disease data sets and to try to infer the results. The resources include a computer used for the analysis, this is crucial in defining the scope of the study because computer used defines how much and how fast data can be analysed. Computer used in this study was enough to perform major tasks but not enough for in-depth and optimal results. The schedule is set as per the thesis requirements, which restrict this study to examine algorithms in greater detail. In the third step, the basic knowledge to get started with the research was defined. For that several online resources were used and fundamentals for the study were understood. In fourth step, data was collected online from machine learning repository UCI, Irvine. In the fifth step, descriptive and exploratory data analysis as well as data cleaning was done as per the knowledge gained from step 3. It helped to understand the data more such as features included, correlation between features, missing values and outliers. In the sixth phase, different literature was studied related to this heart disease data set. The literature was available online and the findings of the literatures were reviewed. In the seventh phase, different machine learning algorithms were trained and the results were obtained as per different performance metrics. In the eighth phase, results were interpreted and compared with other existing literature on the subject. In ninth and final phase, conclusions were derived based on the results obtained.

3.3 Data Collection and Description

The data is collected from UCI machine learning repository. The data set is named Heart Disease Data Set and can be found in UCI machine learning repository. The UCI machine learning repository contains vast and varied amount of datasets which include datasets from various domains. These data are widely used by machine learning community from novices to experts to understand data empirically. Various academic papers and researches have been conducted using this repository. This repository was created in 1987 by David Aha and fellow students at UCI Irvine. [24]
Heart disease data set contains data from four institutions.

1. Cleveland Clinic Foundation.
2. Hungarian Institute of Cardiology, Budapest.
3. V.A. Medical Centre, Long Beach, CA.
4. University Hospital, Zurich, Switzerland.

For the purpose of this study, the data set provided by Cleveland Clinic Foundation is used. This dataset was provided by Robert Detrano, M.D, Ph.D. Reason to choose this dataset is, it has less missing values and is also widely used by research community. [25]

Table 2. Features of the Heart Disease dataset

Features	**Description**
Age	Age in years
Sex	Gender instance (0 = Female, 1 = Male)
Cp	Chest pain type (1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic)
Trestbps	Resting blood pressure in mm Hg
Chol	Serum cholesterol in mg/dl
Fbs	Fasting blood sugar > 120 mg/dl (0 = False, 1= True)
Restecg	Resting ECG results (0: normal, 1: ST-T wave abnormality, 2: LV hypertrophy)
Thalach	Maximum heart rate achieved
Exang	Exercise induced angina (0: No, 1: Yes)
Oldpeak	ST depression induced by exercise relative to rest
Slope	Slope of the peak exercise ST segment (1: up-sloping, 2: flat, 3: down-sloping)
Ca	Number of major vessels coloured by fluoroscopy (values 0 - 3)
Thal	Defect types: value 3: normal, 6: fixed defect, 7: irreversible defect
Num	Diagnosis of heart disease (0: Healthy, 1: Unhealthy)

There are actually 76 attributes in the dataset but only 14 attributes are used for this study, these 14 attributes are in Table 2.

3.4 Tools Used

Different tools are used for this study. All of them are free and open source.

1. Python 3.5
2. NumPy 1.11.3
3. Matplotlib 1.5.3
4. Pandas 0.19.1
5. Seaborn 0.7.1
6. SciPy and Scikit-learn 0.18.1

Python is a high level general programming language and is very widely used in all types of disciplines such as general programming, web development, software development, data analysis, machine learning etc. Python is used for this project because it is very flexible and easy to use and also documentation and community support is very large [26].

NumPy is very powerful package which enables us for scientific computing. It comes with sophisticated functions and is able to perform N-dimensional array, algebra, Fourier transform etc. NumPy is used very where in data analysis, image processing and also different other libraries are built above NumPy and NumPy acts as a base stack for those libraries [27].

Pandas is open source BSD licensed software specially written for python programming language. It provides complete set of data analysis tools for python and is best competitor for R programming language. Operations like reading data-frame, reading csv and excel files, slicing, indexing, merging, handling missing data etc., can be easily performed with Pandas. Most important feature of Pandas is, it can perform time series analysis [28].

Seaborn is library used for data visualization and is created by using python programming language. It is high level library stacked on top of matplotlib. Seaborn is more attractive and informative than matplotlib and very easy to use and is tightly integrated with NumPy and Pandas. Seaborn and matplotlib can be used essentially side by side to derive conclusions from the datasets [29].

SciPy is a collection of fundamental mathematical functions and is built on the top of Numpy while Scikit-learn is widely used popular library for machine learning, it is third party extension to SciPy. Scikit-learn includes all the tools and algorithms needed for most of machine learning tasks. Scikit-learn supports regression, classification, clustering, dimensionality reduction and data pre-processing. For this study, scikit-learn is used because it is based on python and can interoperate to NumPy library. It is also very easy to use [30].

3.5 Data Cleaning and Scaling

The Cleveland dataset used for this study contains 303 rows, 5 numerical and 9 categorical attributes. The data set has 6 missing values. Since the missing values are from categorical attributes it was replaced with mode value of the respective columns. There are 43 outliers in the dataset but it is not a noise or wrong input. Data falling outside the 3 standard deviation are considered outliers. Patients who are diagnosed with heart disease have these outliers, meaning their cholesterol and blood pressure data is above the normal scale. So, in real clinical scenario it is not outlier and therefore it is decided to keep these outliers as they are. Standardization was used to rescale the data. That helped the machine learning models to perform better.

3.6 Data Reduction

For the Cleveland dataset, Sequential Backward Elimination is used. Any features in the data set whose p-value is less than 0.05 is significant and is considered essential in this data set. The logistic model was used as a classifier for sequential backward elimination.

Out of 13 dependent attributes 9 attributes are selected by using sequential backward elimination. Data reduced features selected are listed below.

1. Sex
2. Chest pain type (Cp)
3. Resting blood pressure (Trestbps)
4. Maximum heart rate achieved (Thalach)
5. ST depression induced by exercise (Oldpeak)
6. Slope of the peak exercise ST segment (Slope)
7. Number of major vessels coloured by fluoroscopy (Ca)

3.7 Descriptive Statistics and Exploratory Data Analysis

In this section, both descriptive statistics and exploratory data analysis are discussed. Out of 14 attributes, the data contains only five numeric attributes, out of which only four are tabulated below. Oldpeak attribute is excluded from the table because it needs deep understanding of heart physiology to interpret its results.

Table 3. Descriptive statistics of Heart Disease dataset.

	Age	**Trestbps**	**Chol**	**Thalach**
mean	54.43	131.68	246.69	149.60
std	9.03	17.59	51.77	22.87
min	29.00	94.00	126.00	71.00
median	56.00	130.00	241.00	153.00
mode	58.00	120.00	197.00	162.00
max	77.00	200.00	564.00	202.00

In Table 3, the distribution of data is non-symmetrical and skewed. Population of data is 303, whose age range is 29 to 77. Age distribution of data is left skewed which shows population with low age are absent. Mean, median and mode of age column are 54.43, 56.00 and 58.00 respectively. Standard deviation is 9.03 which shows the sparseness of the age, ranging from 53.40 to 64.46, this age range visits the doctor most. Similarly, Trestbps, blood pressure column's data is right skewed, which shows either population with normal blood pressure is present or there is lack of high blood pressure. Average blood pressure is 131.68 which is slightly above normal blood pressure, 120, this shows population is in pre-high blood pressure zone. Minimum blood pressure is 94 which is higher than low blood pressure 90, this shows population has not low blood pressure problems. In Chol, column mean value is 246.69 which mean population has high cholesterol level, cholesterol border line value is from 200 to 239. Mode value is 197.00 which shows most of the population has desirable cholesterol level, which is less than 200. Cholesterol mean value of 246.69 is abnormal in contrast to mode value of 197. In Thalach, maximum heart rate achieved, mean value is 149.60 with standard deviation of 22.87 with minimum value of 71.00 and maximum value of 202.00 which shows heart beat rate in average is less than normal of 200 beats per minute.

Table 4 shows the correlation of numeric data.

Table 4. Corelation matrix between numeric attributes

	Age	Trestbps	Chol	Thalach
Age	1.00	0.28	0.20	-0.39
Trestbps	0.28	1.00	0.13	-0.04
Chol	0.20	0.13	1.00	-0.003
Thalach	-0.39	-0.04	-0.003	1.00

From Table 4, it can be concluded that age and maximum heart rate achieved has some correlation. As, age increases, heart rate decreases except that none of other attributes shows any high correlation with each other.

From the exploratory analysis of the data, the following insights are gained. Two of the insights are presented in figure 11.

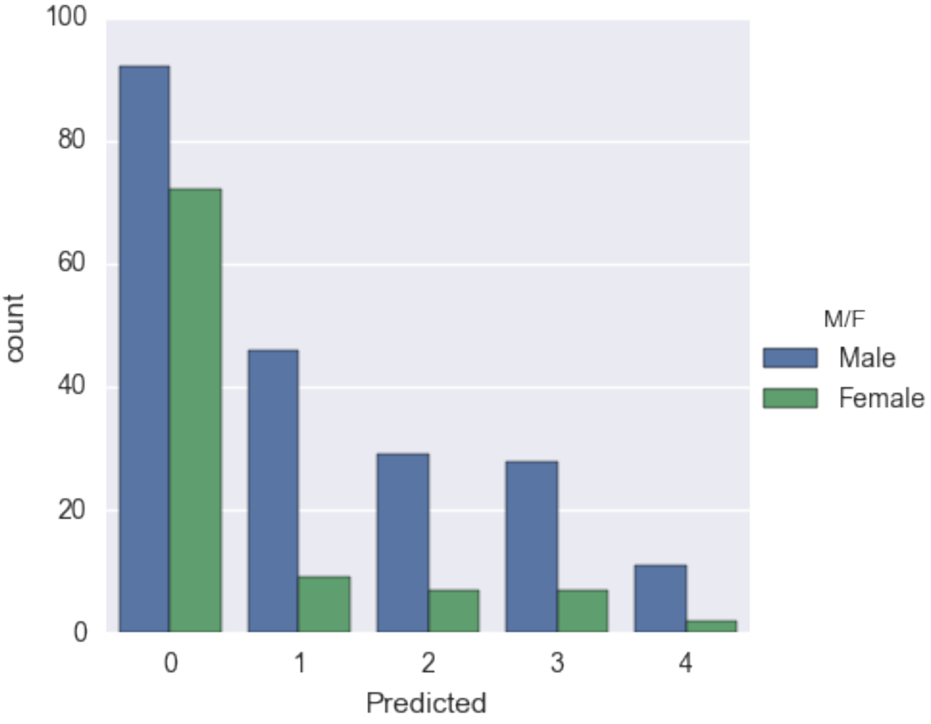


Figure 11 Disease rate in the male and the female.

In figure 11, 0,1,2,3 and 4 are the status of heart health ranging from healthy to severely unhealthy. Blue bar represents male population and green bar represents female population. It can be seen that; in this data set the male population is more prone to heart disease. To verify, chi-square test is performed, for that null hypothesis H_0 is proposed. Null hypothesis states that disease has no correlation with population's gender, if this hypothesis is rejected by chi-square test then, hypothesis Hα is established which states that disease has some correlation with population's gender.

Table 5. Number of diseased patients based on sex.

	0	1	2	3	4	Total
Male	92	46	29	28	11	206
Female	72	9	7	7	2	97
Total	164	55	36	35	13	303

In Table 5, the total number of male population is 206 and that of female population is 97. 0,1,2,3,4 are the columns ranging from, no heart disease, to severely unhealthy heart disease, and its total values are 164, 55, 36, 35 and 13 respectively.

Table 6. Expected number of diseased patients based on sex.

	0	1	2	3	4
Male	111.4	37.39	24.47	23.79	8.83
Female	52.50	17.60	11.52	11.20	4.16

In Table 6, the expected number of diseased patients are calculated. One example of how the expected number is derived for 0 column is as follow:

$$Male = \frac{Total\ number\ of\ Male\ population * Total\ number\ of\ population\ in\ 0\ column}{Total\ number\ of\ whole\ population}$$

$$Male = \frac{206 * 164}{303} = 111.4$$

Formula for calculating chi-square test is,

$$Chi = \sum \frac{(Observed - Expected)^2}{Expected} \quad (3.1)$$

Table 6, contains five columns 0,1,2,3,4 and two rows male and female. Degrees of freedom for this table is (R-1) *(C-1) where R and C are rows and columns respectively. Therefore, degree of freedom is 4.

The value obtained from the chi-square test is 23.35, which is greater than the value from the chi-square distribution table which is 9.488. Therefore, the null hypothesis is rejected. Hence, it can be stated with 95% confidence that disease has some correlation with gender. This concludes, as shown in figure 13, that the male population is more prone to heart disease than female population.

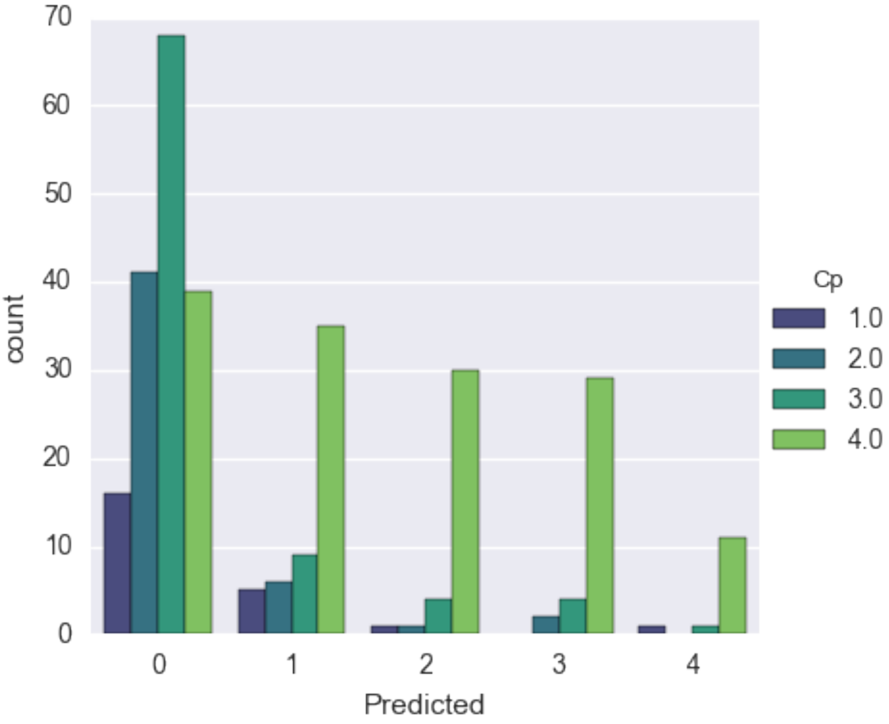


Figure 12 Disease prediction according to chest pain type

In figure 12, in x-axis, there is the predicted value of heart disease ranging 0 to 4 denoting healthy to severely unhealthy. Cp, chest pain type ranges from 0 to 4 denoting (1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic). In this data set, it can be seen that the population diagnosed with chest pain type 4, asymptomatic chest pain is more likely to have heart disease.

4 Results and Discussion

This section deals with the results obtained from the study of Cleveland dataset and also a comparative analysis of algorithms. After all pre-processing, descriptive and exploratory analysis the data set was employed on different machine learning algorithms, also called models.

4.1 Overview

There are two classes found in Scikit-learn machine learning library called LabelEncoder and OneHotEncoder. LabelEncoder basically transforms the categorical values into numbers which are ordinal in nature. In data set used for this study, there are categorical variables such as Cp, chest pain type which is represented as 1,2,3 and 4. 1,2,3 and 4 does not have ordinal relationship with each other therefore it gives wrong results when applied directly to machine learning algorithms. Thus, OneHotEncder is used to encode chest pain type values into binary values, this resolves the issue of ordinality. In this data set the dependent variable or the value to be predicted is multi class. It ranges from 0 to 4. But for this study, multiclass dependent variable is converted into binary class.

4.2 Experiments

Different machine learning models were experimented using Cleveland dataset. In this study initially data set was modelled without feature selection and the results were obtained and in the second phase, data set was modelled only with features obtained from SBS. All experiments included methods like k-fold cross validation and parameter tuning. K-fold cross validation is a technique used in a dataset to avoid over-fitting and under-fitting of the model and parameter tuning is technique which assists to find the best parameters for the model being used.

4.2.1 Training set

Training set is the portion of data in which the model is trained. In this study, 70 percent of data was used for training. In general, in machine learning communities, it is a norm to used 60 to 70 percent of data for training but it varies diversely according to the need and purpose of the experiment. In data training, often the accuracy of training is high, meaning the model shows high level of accuracy performance in the training set but when tested against the test set, the performance is poor. So to avoid performance error, k-fold cross validation was used. In k-fold cross validation, for example 10-fold cross validation, training set is split in 10 parts and and from each 10 part, training and test set is defined and model is employed and the result of all the 10 parts are averaged, this helps to minimize the over fitting and under fitting of the data.

4.2.2 Test set

The test set is the portion of data where the model is tested, it is often the dependent variable of the data. In this study, 30 percent of the data was used for testing. When cross validated data is tested, it will perform better or worse depending on the model used. So, to ensure every model is functioning in its optimum, a technique named parameter tuning was used. Scikit-learn library contains the class called GridSearchCV which performs the parameter tuning.

4.2.3 Results

The aim of the entire project was to test which algorithm classifies diseases the best. This section includes all the results obtained from the study and introduces the best performer according to various performance metrics.

First, performance was obtained by using 10-fold cross validation in training set. Secondly, performance was obtained just by using model without any parameter tuning, third parameters were tuned and fourth model was calibrated. The following tables shows the results.

Table 7. Evaluation of algorithms in training set using k-fold.

Algorithms used	Accuracy	Precision	Recall	F1-score	ROC AUC	Log loss
KNN	0.84	0.89	0.73	0.83	0.90	0.42
SVC	0.82	0.81	0.79	0.82	0.89	0.41
Random Forest	0.67	0.64	0.63	0.66	0.85	11.30
Naïve Bayes	0.80	0.81	0.72	0.79	0.87	1.51

Referring Table 7, KNN gave best results in the 10-fold cross validation followed by SVC, Naïve Bayes and Random Forest.

Following Table, shows the performance of algorithms in test set without parameter tuning.

Table 8. Evaluation of algorithms in test set without parameter tuning.

Algorithms used	Accuracy	Precision	Recall	F1-score	ROC AUC	Log loss
KNN	0.82	0.83	0.81	0.81	0.81	0.44
SVC	0.79	0.78	0.78	0.78	0.78	0.45
Random Forest	0.69	0.68	0.68	0.68	0.69	11.00
Naïve Bayes	0.81	0.81	0.80	0.80	0.80	1.10

Referring to Table 8, KNN is the best performer followed by Naïve Bayes, SVC and Random Forest. In the test set, Naïve Bayes performed better than in the training set but log loss of Naïve Bayes is high than SVC. It can mean that accuracy of Naïve Bayes classifier is not entirely true, classifier is over fitted.

Table 9, shows the performance of algorithms in test set using parameter tuning.

Table 9, Evaluation of algorithms in test set using parameter tuning.

Algorithms used	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC AUC**	**Log loss**
KNN	0.78	0.83	0.78	0.77	0.77	0.54
SVC	0.80	0.82	0.80	0.80	0.79	0.43
Random Forest	0.73	0.74	0.74	0.73	0.73	0.49
Naïve Bayes	Nil	Nil	Nil	Nil	Nil	Nil

In Table 9, Naïve Bayes' row is Nil because it does not need parameter tuning, there are no parameters to be tuned in Naïve Bayes. Thus the result from Table 8 was taken for performance comparison. After parameter tuning, KNN performance goes low. SVC, followed by Naïve Bayes and Random Forest are best performer after parameter tuning. In figure 13, performance of log loss is excluded because it is negative in nature and can exceed one in its value, which is not supported by other performance metrics.

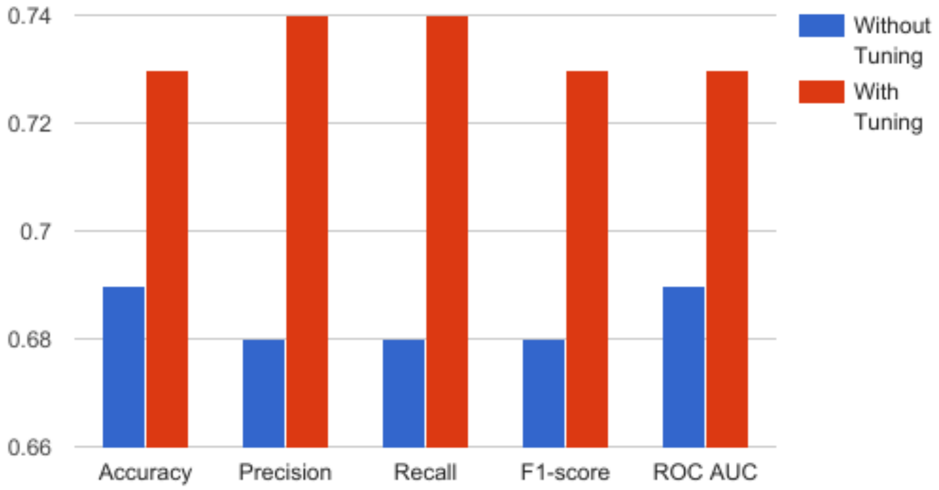


Figure 13 Random Forest with and without parameter tuning

In figure 13, Blue bars represents performance without tuning and red bars represents after tuning. It is seen that after parameter tuning random forest performance is much better than before, still its performance is low in this dataset.

Now all of these algorithms are calibrated using Platt's scaling or also known as sigmoid function. Table 10, shows the results after the calibration.

Table 10, Evaluation of algorithms in test set, parameter tuned using calibration

Algorithms used	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC AUC**	**Log loss**
KNN	0.78	0.78	0.78	0.78	0.77	0.46
SVC	0.79	0.80	0.79	0.79	0.78	0.43
Random Forest	0.75	0.73	0.73	0.72	0.72	0.5
Naïve Bayes	0.80	0.81	0.80	0.80	0.79	0.48

In Table 10, after calibration Naïve Bayes outperforms all the algorithms and also log loss of Naïve Bayes has decreased from 1.10 to 0.48. SVC, KNN and Random Forest are the best models respectively.

Table 11, shows the performance of algorithms after they were tuned and calibrated and feature selected using SBS. This helps to compare the model performance before and after feature selection.

Table 11. Evaluation of tuned and calibrated algorithms with feature selection.

Algorithms used	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC AUC**	**Log loss**
KNN	0.78	0.78	0.78	0.78	0.77	0.42
SVC	0.85	0.86	0.86	0.86	0.85	0.37
Random Forest	0.78	0.78	0.78	0.78	0.77	0.45
Naïve Bayes	0.79	0.80	0.79	0.79	0.78	0.92

Referring to Table 11, SVC is the best classifier followed by Naïve Bayes, KNN and Random Forest.

Table 12, shows the performance of algorithms after they were boosted with tuned and calibrated parameters. SVC, Random Forest and Naïve Bayes supports the Adaboosting while KNN does not.

Table 12, Evaluation of boosted algorithms in test set after without feature selection.

Algorithms used	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC AUC**	**Log loss**
SVC	0.78	0.79	0.78	0.78	0.77	0.43
Random Forest	0.72	0.73	0.73	0.72	0.72	0.5
Naïve Bayes	0.72	0.73	0.73	0.72	0.72	0.49

After applying adaboosting in Table 12, SVC and Naïve Bayes shows the low performance relative to Table 10. Random Forest and Naïve Bayes are almost equal in their performance while SVC outperforms others. KNN which is not supported by Adaboost, performs almost equal to SVC.

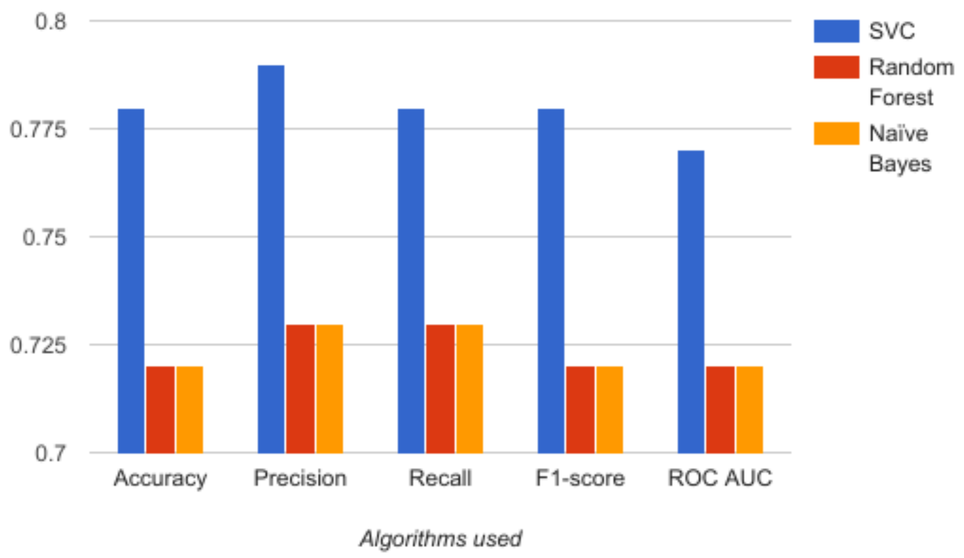


Figure 14 Boosted Algorithms without feature selection.

Table 13, shows the boosted, feature selected, tuned and calibrated algorithms.

Table 13, Evaluation of boosted, tuned and calibrated algorithms with feature selection.

Algorithms used	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC AUC**	**Log loss**
SVC	0.82	0.83	0.82	0.82	0.82	0.38
Random Forest	0.80	0.80	0.80	0.80	0.80	0.5
Naïve Bayes	0.83	0.84	0.84	0.83	0.83	0.43

After Adaboot is applied, Naïve Bayes outperforms other algorithms still its log loss is high, if log loss is taken into account then SVC outperforms all other algorithms. Random Forest also performs better relatively after Adaboosting.

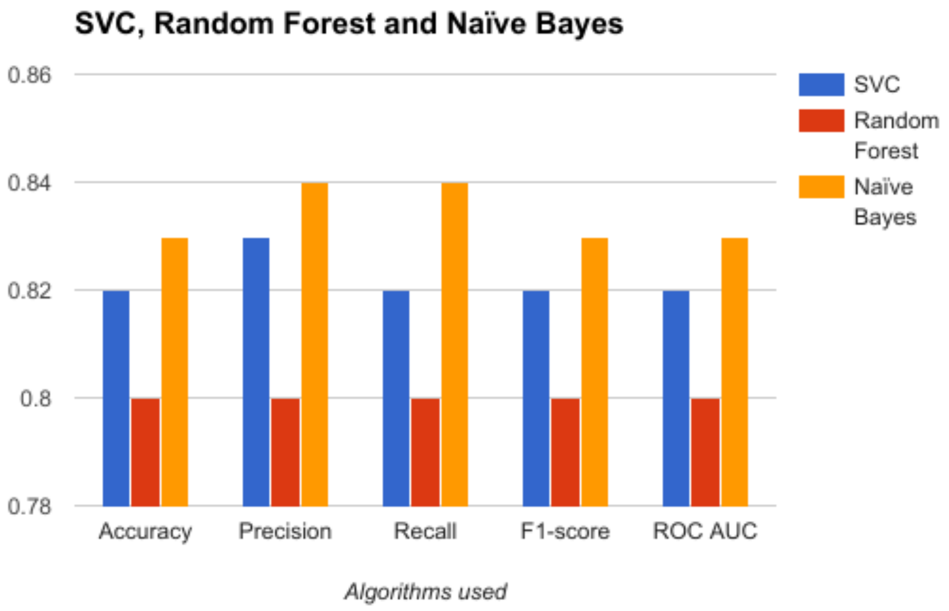


Figure 15 Boosted algorithms with feature selection

From the comparison of figure 14 and 15, Naïve Bayes performance increases significantly after feature selection and outperforms other algorithms if log loss is not taken into account.

Artificial Neural Network(ANN) is another algorithm used in this study. Performance metrics of ANN is in Table 11.

Table 14, Evaluation of Artificial Neural Network

Algorithms used	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC AUC**	**Log loss**
ANN	0.91	0.92	0.89	0.90	0.87	0.23

ANN in Table 14, uses 3 hidden layers and also uses ‘adam’ algorithm as an optimizer. ANN was parameter tuned with Keras machine learning library.

4.3 Comparison of Results

From all the tables above, different algorithms performed better depending upon the situation whether cross validation, grid search, calibration and feature selection is used or not. Every algorithm has its intrinsic capacity to outperform other algorithm depending upon the situation. For example, Random Forest performs much better with a large number of datasets than when data is small while Support Vector Machine performs better with a smaller number of data sets.

Performance of algorithms decreased after boosting in the data which was not feature selected while algorithms were performing better without boosting in not feature selected data. The performance of algorithms increased after boosting in the data which was feature selected while algorithm performance decreased after boosting in feature selected data. This shows the necessity that the data should be feature selected before applying boosting. For the comparison of dataset, performance metrics after feature selection, parameter tuning and calibration is used because this is standard process of evaluating algorithms.

In Table 13, performance metrics of classifiers are added except log loss because lower the log loss better is the classifier, so log loss is subtracted from the added performance metrics and then averaged. Average value of SVC, Random Forest and Naïve Bayes are 0.63, 0.59 and 0.63 respectively. This shows SVC and Naïve Bayes are performing on average.

In Table 14, ANN has highest classification accuracy. From the comparison of SVC or Naïve Bayes with ANN, ANN is the best classifier with 0.71 average value. But ANN has a problem when applied in a clinical environment, because there is less space for the interpretation of the results. ANN works like a black-box, so it cannot be completely trusted before actually knowing how it is functioning.

5 Conclusion

The goal of the project was to compare the algorithms with different performance metrics using machine learning. All data were pre-processed and used for the prediction on the test. Every algorithm performed better in some situation and worse in an other. SVC, and ANN and Naïve Bayes are the likely models to work best in the dataset used in this study.

To be able to diagnose the heart disease accurately using machine learning has many significances. Different devices can be manufactured which will monitor the heart related activities and diagnose the disease. These devices will prove to be helpful where heart disease experts are not available. With further research machine learning can also be used to diagnose the heart disease before the human experts can do.

One of the main achievements of this project was that project helped to understand the algorithms better. When tested through various situations, the algorithms performed differently which helped me to understand the algorithm’s working mechanism. This thesis can be first learning step in heart disease diagnosis with machine learning and it can be extended further for future research. There are several limitations to this study primarily the knowledge base of the author, secondly, the tools used in this study such as processing power of the computer and thirdly the time limitation available for the study. This type of study requires state-of-art resources and expertise in respective domains.

References

1. The Atlas of Heart Disease and Stroke [online].
   URL: http://www.who.int/cardiovascular_diseases/resources/atlas/en/
   Accessed 26 February 2017.

2. CSC Infographic Big Data [online].
   URL:http://assets1.csc.com/insights/downloads/CSC_Infographic_Big_Data.pdf
   Accessed 26 February 2017.

3. Machine Learning Techniques for Multimedia Case Studies on Organization and Retrieval Cord, M.: Cunningham, P, (Eds) 2008, XVI, 289 p, Hardcover

4. Business Dictionary [online].
   URL: http://www.businessdictionary.com/definition/regression.html
   Accessed 26 February 2017.

5. Max Bramer, Principles of Data Mining. 2nd ed. Springer; 2013

6. Robert A Wilson. The MIT Encyclopaedia of Cognitive Sciences. MIT Press; 1999

7. Ethan Alapydin. Introduction to Machine Learning, 2nd ed. Cambridge Massachusetts, MIT Press:2010.

8. Md Rahat Hossain, The Combined Effect of Applying Feature Selection and Parameter Optimzation on Machine Learning Techniques for Solar Power Prediction. American Journal of Energy Research 2013; 1(1): 7-16.

9. Gareth James. Introduction to Statistical Learning. New York, Springer; 2013.

10. Ömer Cengiz Çelebi. Principal Component Analysis [online]. 26 February 2002
    URL:http://www.byclb.com/TR/Tutorials/neural_networks/ch5_1.htm
    Accessed 26 February 2017.

11. Lindasay I Smith. A tutorial on Principal Component Analysis [online]. URL: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf Accessed 8 March 2017.

12. Laetitia Van Cauwenberge. Top 10 Machine Learning Algorithms [online].Data Science Central; 6 December 2015 URL:http://www.datasciencecentral.com/profiles/blogs/top-10-machine-learning-algorithms Accessed 26 February 2017.

13. Shai Shalev-Shwartz, Shai Ben-David. Understanding Machine Learning. New York, Cambridge University Press; 2014.

14. Michael Luk. K-Nearest Neighbour and Dynamic Time Wrapping [online]. Devices using DTW and KNN; 16 July 2016 URL:https://sflscientific.com/case-studies/2016/6/4/time-series-analysis-fitbit-using-dtw-and-knn Accessed 26 February 2017.

15. Introduction to Support Vector Machines [online]. Open-CV Documentation; 7 March 2017 URL: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html Accessed 26 February 2017.

16. Robert Scaphire. Explaining AdaBoost [online]. URL:http://rob.schapire.net/papers/explaining-adaboost.pdf Accessed 26 February 2017.

17. Hastie, Tibshirani, Friedman. The Elements of Statistical Learning. 2nd ed. Springer Series in Statistics; 2009

18. Computer Science Department Darmouth College. A Basic Introduction to Neural Networks [online].
URL:
https://www.cs.duke.edu/brd/Teaching/Previous/AI/Lectures/NN/neural.html
Accessed 26 February 2017

19. Thomas G. Tape.The Area Under ROC Curve [online]. University of Nebraska Medical Center; 17 December 2016
URL: http://gim.unmc.edu/dxtests/roc3.htm
Accessed 26 February 2017.

20. Log Loss [online].
URL:
http://scikitlearn.org/stable/modules/generated/sklearn.metrics.log_loss.html
Accessed 26 February 2017.

21. Probability Calibration [online].
URL: http://scikit-learn.org/stable/modules/calibration.html
Accessed 26 February 2017

22. Yin. Case Study Research: Design and Methods (Applied Social Research Methods). 5th edition. Los Angeles, Sage Publications; 2013

23. Burns N, Grove K. The Practice of Nursing Research: Conduct, Critique and Utilization. 5th ed. St. Louis, Elsevier Saunders; 2005

24. UCI Machine Learning Repository [online].
URL: http://archive.ics.uci.edu/ml/about.html
Accessed 26 February 2017.

25. UCI Machine Learning Repository [online].
URL:https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/heart-disease.names
Accessed 26 February 2017.

26. Python Programming Documentation [online].

    URL: https://www.python.org/about/

    Accessed 26 February 2017.

27. Numpy Documentation [online].

    URL: http://www.numpy.org/

    Accessed 26 February 2017.

28. Pandas Documentation [online].

    URL :http://pandas.pydata.org/

    Accessed 26 February 2017

29. Michael Waksom. An Introduction to Seaborn [online].

    URL :http://seaborn.pydata.org/introduction.html

    Accessed 26 February 2017

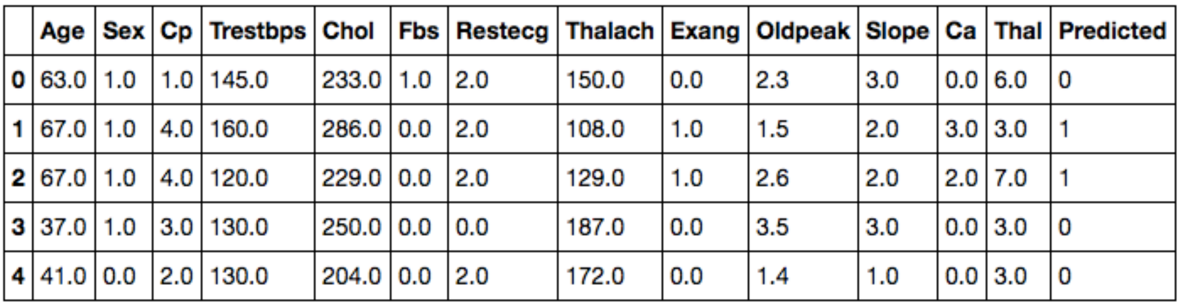
30. Fabian Pedregosa. Scikit-learn: Machine Learning in Python [online].

    URL: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

    Accessed 26 February 2017

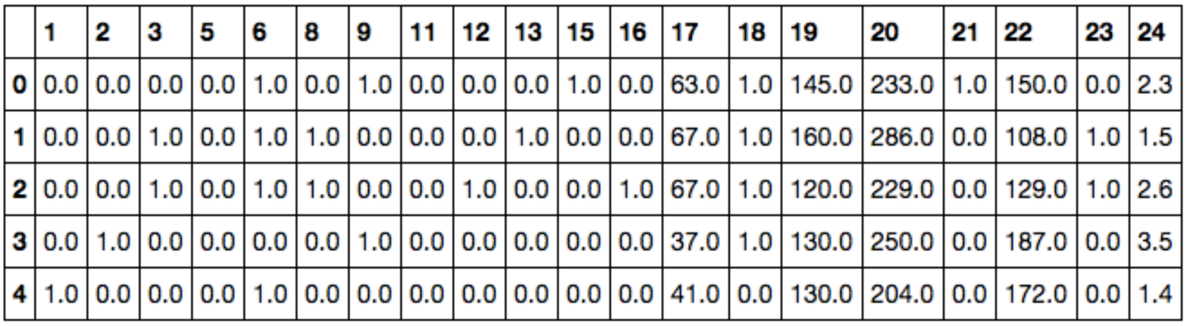
Appendixes

Appendix 1. Sample of Data



	Age	Sex	Cp	Trestbps	Chol	Fbs	Restecg	Thalach	Exang	Oldpeak	Slope	Ca	Thal	Predicted
0	63.0	1.0	1.0	145.0	233.0	1.0	2.0	150.0	0.0	2.3	3.0	0.0	6.0	0
1	67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	2.0	3.0	3.0	1
2	67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	2.0	2.0	7.0	1
3	37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.0	3.0	0
4	41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4	1.0	0.0	3.0	0

Appendix 2. Sample of Onehotencoded Data



	1	2	3	5	6	8	9	11	12	13	15	16	17	18	19	20	21	22	23	24
0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	63.0	1.0	145.0	233.0	1.0	150.0	0.0	2.3
1	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	67.0	1.0	160.0	286.0	0.0	108.0	1.0	1.5
2	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	67.0	1.0	120.0	229.0	0.0	129.0	1.0	2.6
3	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	37.0	1.0	130.0	250.0	0.0	187.0	0.0	3.5
4	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	41.0	0.0	130.0	204.0	0.0	172.0	0.0	1.4

Appendix 3. Sample code of Naïve Bayes with Feature Selection

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# reading data
df = pd.read_csv('processed.cleveland.csv', names= ['Age',
'Sex', 'Cp','Trestbps','Chol','Fbs','Restecg',
'Thalach','Exang','Oldpeak','Slope','Ca','Thal','Predicted'])

# since data is in string it needs to changed to numeric
```

```
df = df.convert_objects(convert_numeric= True)

# Missing categorical values are filled with mode.
df['Ca'] = df['Ca'].fillna(value = df['Ca'].mode()[0])
df['Thal'] = df['Thal'].fillna(value =
df['Thal'].mode()[0])

# Since predicted column is multilabel, it is changed to
binary.
df['Predicted'] = (df['Predicted'] > 0).astype(int)

# These columns are dropped because they are insignificant,
results got from
# sequential backward elimination

df = df.drop(['Age','Chol','Fbs','Restecg','Exang','Thal'],
axis= 1)

# X is independent variable, y is dependent variable to be
predicted
X = df.iloc[:, :-1].values
y = df.iloc[:, 7].values

# Ordinal data is encoded
from sklearn.preprocessing import LabelEncoder,OneHotEncoder
onehotencoder = OneHotEncoder(categorical_features=[1,5,6])
X = onehotencoder.fit_transform(X).toarray()

X = pd.DataFrame(X)
X = X.drop(X.columns[[0,4,7]],axis=1)

# Data is standardized
```

```
from sklearn.preprocessing import StandardScaler
scaler =  StandardScaler()
scaler.fit(X)
scaled_features = scaler.transform(X)
df_feat = pd.DataFrame(scaled_features)

X = df_feat
y = df['Predicted']

# Data is splitted into training set and test test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 0)
from sklearn.naive_bayes import GaussianNB
naive = GaussianNB()           # gaussian naive bayes classifier is called.
naive.fit(X_train, y_train)
predictions = naive.predict(X_test)

# Modules for calibration, performance metrics are imported

from sklearn.calibration import CalibratedClassifierCV
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import accuracy_score, roc_auc_score, roc_curve, log_loss,f1_score
cal_cv = CalibratedClassifierCV(base_estimator=naive,method='sigmoid', cv = 10)
cal_cv.fit(X_train, y_train)
calibrated_prediction = cal_cv.predict(X_test)
print(confusion_matrix(y_test, calibrated_prediction))
print(classification_report(y_test, calibrated_prediction))
print('Accuracy', accuracy_score(y_test, calibrated_prediction))
print('ROC', roc_auc_score(y_test, calibrated_prediction))
calibrated_proba = cal_cv.predict_proba(X_test)
print("loss WITH calibration : ", log_loss(y_test, calibrated_proba, eps=1e-15, normalize=True
```